

1. A branch control apparatus in a microprocessor,
comprising:

an instruction cache, for outputting a line of
instruction bytes selected by a fetch address;

an instruction buffer, coupled to said instruction
cache, for buffering said line of instruction
bytes;

a branch target address cache (BTAC), coupled to said
fetch address, for providing offset information
relating to a location of a branch instruction
within said line of instruction bytes; and

selection logic, coupled to said BTAC, for causing a
portion of said instruction bytes to not be
provided to said instruction buffer, based on
said offset information.
2. The apparatus of claim 1, wherein said offset
information specifies a location of an instruction
immediately following said branch instruction within
said line of instruction bytes.
3. The apparatus of claim 2, wherein said portion of said
instruction bytes not provided to said instruction

buffer comprises instruction bytes immediately following said branch instruction within said line of instruction bytes as specified by said offset.

4. The apparatus of claim 1, wherein said selection logic comprises:

a register, coupled between said instruction cache and said instruction buffer, for storing said line of instruction bytes.

5. The apparatus of claim 4, wherein said selection logic further comprises:

a plurality of valid bits, coupled to said register, wherein each of said plurality of valid bits is associated with one of said instruction bytes in said register.

6. The apparatus of claim 5, wherein said selection logic populates said plurality of valid bits based on said offset information received from said BTAC.

7. The apparatus of claim 6, wherein said selection logic causes each of said instruction bytes in said register having a corresponding valid bit that indicates said

corresponding instruction byte is invalid to not be provided to said instruction buffer.

8. The apparatus of claim 7, wherein said BTAC provides a hit signal to said selection logic for indicating whether or not said fetch address hit in said BTAC.
9. The apparatus of claim 8, wherein said selection logic populates said plurality of valid bits based on said offset information received from said BTAC if said hit signal indicates said fetch address hit in said BTAC.
10. The apparatus of claim 5, wherein said selection logic comprises:

muxing logic, coupled between said instruction cache

and said instruction buffer, for causing ones of

said instruction bytes indicated as valid by said

associated valid bit to be provided to said

instruction buffer.
11. The apparatus of claim 10, wherein said muxing logic comprises a set of muxes for discarding ones of said instruction bytes indicated as invalid by said associated valid bit.

12. The apparatus of claim 10, wherein said muxing logic comprises a set of muxes for aligning ones of said instruction bytes indicated as valid by said associated valid bit with a first empty location in said instruction buffer.
13. The apparatus of claim 10, wherein said muxing logic comprises a set of muxes for shifting ones of said instruction bytes indicated as valid by said associated valid bit by a number of bytes shifted out of said instruction buffer.
14. The apparatus of claim 13, wherein said selection logic is configured to receive a shift count from instruction format logic for indicating a number of instruction bytes to be shifted out of said instruction buffer.
15. The apparatus of claim 14, wherein said muxing logic shifts said ones of said instruction bytes indicated as valid by said associated valid bit by said shift count.
16. The apparatus of claim 1, wherein said instruction buffer comprises a shift register.

17. The apparatus of claim 16, wherein said shift register is a byte-wide.
18. The apparatus of claim 1, wherein said instruction buffer is directly coupled to instruction format logic that formats said instruction bytes.
19. The apparatus of claim 18, wherein a bottom byte of said instruction buffer is provided directly to a portion of said instruction format logic configured to a first byte of an instruction for formatting.
20. The apparatus of claim 1, wherein said branch instruction comprises an x86 branch instruction.
21. The apparatus of claim 1, wherein said BTAC is configured to provide a target address of said branch instruction in response to said fetch address.
22. The apparatus of claim 21, wherein said target address is selectively provided to said instruction cache as a subsequent fetch address for selecting a second line of instruction bytes containing a target instruction of said branch instruction in said instruction cache.
23. The apparatus of claim 22, wherein said selection logic causes said target instruction to be provided to

said instruction buffer adjacent to said branch
instruction within said instruction buffer.

24. The apparatus of claim 23, wherein said selection
logic causes instruction bytes preceding said target
instruction in said second line to be discarded and
not provided to said instruction buffer.
25. The apparatus of claim 1, wherein said instruction
cache stores variable length instructions for
execution by the microprocessor.

26. A pre-decode stage within a microprocessor,
comprising:

an instruction buffer, for buffering instruction data
for provision to instruction format logic;

selection logic, coupled to said instruction buffer,
for receiving first instruction data selected by
a fetch address from an instruction cache, said
first instruction data including a branch
instruction; and

a branch target address cache (BTAC), coupled to said
selection logic, for providing a target address
of said branch instruction as a next fetch
address to said instruction cache;

wherein said selection logic is configured to receive
second instruction data selected by said target
address from said instruction cache, said second
instruction data including a target instruction
of said branch instruction; and

wherein said selection logic is configured to write
said branch instruction and said target
instruction immediately adjacent to one another
into said instruction buffer.

27. The pre-decode stage of claim 26, wherein said BTAC is configured to provide said target address in response to said fetch address.
28. The pre-decode stage of claim 26, wherein said BTAC is configured to provide to said selection logic an indication of a location in said first instruction data that immediately follows said branch instruction.
29. The pre-decode stage of claim 28, wherein said selection logic writes said branch instruction and said target instruction immediately adjacent to one another based on said indication of said location.
30. The pre-decode stage of claim 29, wherein said selection logic is configured to receive said target address.
31. The pre-decode stage of claim 30, wherein said selection logic writes said branch instruction and said target instruction immediately adjacent to one another based on said target address and said indication of said location.
32. The pre-decode stage of claim 28, wherein said BTAC is configured to provide said indication in response to said fetch address.

33. The pre-decode stage of claim 26, wherein said instruction buffer comprises a shift register.
34. The pre-decode stage of claim 33, wherein said shift register is a byte-wide.
35. The pre-decode stage of claim 33, wherein said selection logic writes said branch instruction and said target instruction immediately adjacent to a last valid data byte in said instruction buffer.
36. The pre-decode stage of claim 33, wherein said selection logic writes said branch instruction and said target instruction to a next empty location in said instruction buffer.
37. The pre-decode stage of claim 33, wherein said instruction buffer is directly coupled to said instruction format logic.

38. A method for providing a branch instruction and a target instruction of the branch instruction to an instruction buffer, the method comprising:
- receiving from the instruction cache a first cache line containing the branch instruction;
- receiving from a branch target address cache (BTAC) an offset within said first cache line of an instruction immediately following the branch instruction;
- receiving from the instruction cache a second cache line containing the target instruction, said second cache line selected by a target address of the branch instruction provided by said BTAC;
- discarding instructions after the branch instruction in said first cache line;
- discarding instructions preceding the target instruction in said second cache line; and
- providing to the instruction buffer a portion of said first and second cache lines remaining after each of said discardings.

39. The method of claim 38, wherein said discarding instructions after the branch instruction in said first cache line is performed based on said offset.
40. The method of claim 38, wherein said discarding instructions preceding the target instruction in said second cache line is performed based on said target address.
41. The method of claim 38, further comprising:
- providing a fetch address to the instruction cache prior to said receiving from the instruction cache said first cache line;
- wherein the instruction cache provides said first cache line in response to said fetch address.
42. The method of claim 41, further comprising:
- providing said fetch address to the BTAC prior to said receiving from said BTAC said offset;
- wherein said BTAC provides said offset in response to said fetch address.
43. The method of claim 38, further comprising:

storing said first cache line in a register prior to
said discarding instructions after the branch
instruction in said first cache line.

44. The method of claim 43, wherein said discarding
instructions after the branch instruction in said
first cache line comprises marking said instructions
after the branch instruction in said register invalid,
and not providing said instructions marked invalid in
said register to the instruction buffer.

45. The method of claim 38, further comprising:

storing said second cache line in a register prior to
said discarding instructions preceding the target
instruction in said second cache line.

46. The method of claim 45, wherein said discarding
instructions preceding the target instruction in said
second cache line comprises marking said instructions
preceding the target instruction in said register
invalid, and not providing said instructions marked
invalid to the instruction buffer.